Subject: Re: The issues for agreeing on a virtualization/namespaces implementation.
Posted by ebiederm on Wed, 08 Feb 2006 17:46:14 GMT
View Forum Message <> Reply to Message

Hubertus Franke <frankeh@watson.ibm.com> writes:

>
> So it seems the clone( flags ) is a reasonable approach to create new
> namespaces. Question is what is the initial state of each namespace?
> In pidspace we know we should be creating an empty pidmap !
> In network, someone suggested creating a loopback device
> In uts, create "localhost"
"localhost" is wrong.  Either copy or set it to the value from
system initialization time.

> Are there examples where we rather inherit ?  Filesystem ?
> Can we iterate the assumption for each subsystem what people thing is right?

I think this needs to happen on a pure subsystem basis as we merge the
patches.

> IMHO, there is only a need to refer to a namespace from the global context.
> Since one will be moving into a new container, but getting out of one
> could be prohibitive (e.g. after migration)
> It does not make sense therefore to know the name of a namespace in
> a different container.
>
> The example you used below by using the pid comes natural, because
> that already limits visibility.
>
> I am still struggling with why we need new sys_calls.
> sys_calls already exist for changing certain system parameters (e.g. utsname )
> so to me it boils down to identifying a proper initial state when the
> namespace is created.

Agreed.  But we can't always count on everything having a useful state
that we can modify from the inside.  So it is important to leave the
option open at least for those case.

And again there is always the idea that by adding flags we will
transform fork or fork+exec into a spaghetti system call.  I think
that is a reasonable concern.

Also there is always the danger that we run out of clone flags.

>> What I have done which seems easier than creating new names is to refer
>> to the process which has the namespace I want to manipulate.

>
> Is then the idea to only allow the container->init to manipulate
> or is there need to allow other priviliged processes to perform namespace
> manipulation?
> Also after thinking about it.. why is there a need to have an external name
> for a namespace ?

Largely it connects to the super chroot usage where you have one
sysadmin he has multiple daemons each running in their own environment
for isolation purposes. Nothing is installed in the chroot so an
attacker that gets in cannot do anything.

>>>>6) How do we do all of this efficiently without a noticeable impact on
>>>> performance?
>>>> - I have already heard concerns that I might be introducing cache
>>>> line bounces and thus increasing tasklist_lock hold time.
>>>> Which on big way systems can be a problem.
>>>
>>>Possible to split the lock up now.. one for each pidspace ?
>> At the moment it is worth thinking about. If the problem isn't
>> so bad that people aren't actively working on it we don't have to
>> solve the problem for a little while, just be aware of it.
>>
>
> Agree, just need to be sure we can split it up. But you already keep
> a task list per pid-namespace, so there should be no problem IMHO.
> If so let's do it now and take it of the table it its as simple as
>
> task_list_lock ::= pspace->task_list_lock

Actually I don't although that could be trivial. But it is the
wrong split. The problem is that it is a lock with global effect.

Eric